AD-A258 046

TATION PAGE

Form Approved
OBM No. 0704-0188

...ge 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and ...rmation. Send comments regarding this burden or any other aspect of this collection of information, including suggestions ... for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to ...0188), Washington, DC 20503.

1. Agency Use Only *(Leave blank).*

2. Report Date. Sept.-Oct. 1992

3. Report Type and Dates Covered. Final - Journal Article

4. Title and Subtitle.
Numeric precision in FORTRAN computing

5. Funding Numbers.
Contract
Program Element No. 0602435N
Project No. RJ035
Task No. I21
Accession No. DN496433
Work Unit No. 12420A

6. Author(s).
Roger W. Meredith

7. Performing Organization Name(s) and Address(es).
Naval Oceanographic and Atmospheric Research Laboratory
Ocean Acoustics and Technology Directorate
Stennis Space Center, MS 39529-5004

8. Performing Organization Report Number.
JA 242:005:91

9. Sponsoring/Monitoring Agency Name(s) and Address(es).
Naval Oceanographic and Atmospheric Research Laboratory
Ocean Acoustics and Technology Directorate
Stennis Space Center, MS 39529-5004

10. Sponsoring/Monitoring Agency Report Number.
JA 242:005:91

11. Supplementary Notes.
*Published in Computers in Physics

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*
This article compares both real and complex outputs from sizeable numeric computations using identical code on several computer systems. The digital signal processing technique known as the modified covariance method was used as the computational engine. It is a recursive algorithm for solving the covariance equations of a linear predictor that seeks to predict an input signal by a linear combination of past signal samples. Single precision and double precision results are presented but the study focuses primarily on differences between the VAX Fortran 4.8 and MacFortran/020 compilers. Differences in the first digit for single precision arithmetic were found and double precision differences occurred in the eighth digit. Arithmetic with complex data types was found to be less precise than with real data types. Although differences exist among various computer systems, they all show the same order of magnitude accuracy with respect to CRAY-YMP results. The algorithm used here required a double precision implementation to obtain agreement between different computer systems.

14. Subject Terms.
Acoustic propagation, sea ice, noise

15. Number of Pages. 7

16. Price Code.

17. Security Classification of Report.	18. Security Classification of This Page.	19. Security Classification of Abstract.	20. Limitation of Abstract.
Unclassified	Unclassified	Unclassified	SAR

Numeric precision in FORTRAN computing

Roger W. Meredith
NAVAL Research Laboratory-Stennis Space Center Detachment, Arctic Acoustics Branch, Code 242, Stennis Space Center, Mississippi 39529

(Received 23 October 1990; accepted for publication 13 April 1992)

This article compares both real and complex outputs from sizeable numeric computations using identical code on several computer systems. The digital signal processing technique known as the modified covariance method was used as the computational engine. It is a recursive algorithm for solving the covariance equations of a linear predictor that seeks to predict an input signal by a linear combination of past signal samples. Single precision and double precision results are presented but the study focuses primarily on differences between the VAX Fortran 4.8 and MacFortran/020 compilers. Differences in the first digit for single precision arithmetic were found and double precision differences occurred in the eighth digit. Arithmetic with complex data types was found to be less precise than with real data types. Although differences exist among various computer systems, they all show the same order of magnitude accuracy with respect to CRAY-YMP results. The algorithm used here required a double precision implementation to obtain agreement between different computer systems.

INTRODUCTION

Presently, personal computers have the speed and numerical precision of mainframe computers from a few years ago. For example, the Macintosh SE030 that I use is a full 32-bit machine exploiting a Motorola 68030 CPU, a 68882 floating point coprocessor, and a paged memory management unit. The data bus is 32-bits and the reference clock speed is 16 (MHz). Because of significant increases in desktop capability, numerically intensive FORTRAN codes are now being transported from mainframe to desktop computers.

This article compares single precision and double precision outputs from identical code on several computer systems but focuses primarily on differences between VAX Fortran 4.8 and MacFortran/020. The task that led to this investigation was the implementation of digital signal processing programs on a Macintosh. I wanted a research tool that would perform complex transforms on data using a Fourier transform method, an autoregressive method, and a linear predictor method. These methods are commonly used, well documented, and FORTRAN subroutines are readily available. The linear predictor solution named the modified covariance method was chosen from the book by Marple.[1] The implementation of this code using MacFortran/020 did not match the book's example output. However, the same code implemented in VAX Fortran 4.8 did match. This was surprising since past experience with codes executed both in VAX Fortran and in MacFortran/020 generally agreed to 4–6 digit accuracy.

For the example discussed herein, the single precision output differences are large and show that output from the same code on different computer systems does not always agree. This example indicates that computer codes need careful implementation and extensive testing beyond just getting the example results. Examples with differences in the first digit for single precision arithmetic are given and double precision differences in the eighth digit occur. Although this article explores differences in output between computer systems, the results reported here apply only for this single example. These results are not necessarily typical of other numerical examples or even different input data for this subroutine.

I. THE MODIFIED COVARIANCE METHOD

The modified covariance method was chosen to compare numerical results from the different compilers. This method, a recursive algorithm for solving the modified covariance equations of a linear predictor, is explained in detail by Marple.[1] A linear predictor seeks to predict the input signal by a linear combination of past signal samples. The example in the book employs a model of order 15 for the number of recursions and the input data are the 64-element complex data series, given in the book's Appendix. The output from this code is a real number and a complex array of length equal to the model order. For the book example, the routine performs over 2300 multiplications and additions. For each recursion, the real variable and all elements of the complex array are computed and there are no direct calls to transcendental functions.

Since the book code utilized single precision arithmetic, minor modifications were necessary to convert data types and arithmetic to double precision real and double precision complex. What is reported here is differences and

421 485

relative errors of each computer system with the CRAY results for single precision example and with the VAX results for the double precision example.

II. READING SEQUENTIAL DATA FILES

Read (*,*) is a convenient method for reading data from a sequential file when the field format is unknown. This list-directed read is implemented differently in the VAX Fortran 4.8 and MacFortran/020 compilers. This can cause processing differences because it affects the number of significant digits in the data. Generally, MacFortran/020 will read six significant digits while VAX Fortran reads seven. To eliminate this difference and focus on numerical computation, the data were read with F7.5 format. Still, differences in the input data remain. Table I is the F7.5 read from both systems. For brevity, just the real parts of the complex input data are shown. The values have nine decimal places but are read using an F7.5 format and printed out using the list directed format. VAX Fortran manipulates the precision of certain input values. This I/O difference does not occur when reading into double precision numbers since both compilers read 16 digits. VAX Fortran still manipulates precision if all 16 digits are not filled. The VAX employs other methods to control computation errors as well. For example, conversions to other data types are not always what one might expect. The real number 0.3333333 when converted to double precision goes to 0.33333331314651184 not 0.3333333000000000 or 0.33333333333333332.

TABLE I. Comparison of data input in F7.5 format.

MacFortran/020		VAX Fortran 4.8	
1.349	0.404	1.349	0.404
– 2.117	1.293	– 2.117	1.293
– 1.786	– 0.119	– 1.786	– 0.119
1.162	– 0.522	1.162	– 0.522
1.641	– 0.974	1.641	– 0.974
0.072	0.275	0.071999997	0.275
– 1.564	0.854	– 1.564	0.854
– 1.08	0.289	– 1.08	0.289
0.927	– 0.283	0.927	– 0.283
1.891	– 0.359	1.891	– 0.359
– 0.105	0.102	– 0.105	0.102
– 1.618	– 0.009	– 1.618	– 0.0089999996
– 0.945	0.185	– 0.945	0.185
1.135	– 0.243	1.135	– 0.243
1.855	– 0.27	1.855	– 0.27
– 1.032	0.399	– 1.032	0.399
– 1.571	– 0.25	– 1.571	– 0.25
– 0.243	0.419	– 0.243	0.419
0.838	– 0.05	0.838	– 0.050000001
1.516	– 0.395	1.516	– 0.395
0.257	0.746	– 0.257	0.746
– 2.057	– 0.559	2.057	– 0.559
– 0.578	– 0.344	– 0.578	– 0.344
1.584	0.733	1.584	0.733
0.614	– 0.48	0.614	– 0.48
– 0.71	0.033	– 0.71	0.033
– 1.1	– 0.321	– 1.1	– 0.321
0.15	– 0.063	0.15	– 0.063000001
0.748	1.239	0.748	1.239
0.795	0.083	0.795	0.082999997
– 0.071	– 0.762	– 0.071000002	– 0.762
– 1.732	– 0.895	– 1.732	– 0.895

III. DATA TYPES

The VAX Fortran 4.8 compiler[2] claims seven digit precision for single precision real numbers. The double precision default is the D – floating option that claims 16-digit precision. The G – floating option claims 15-digit accuracy but has extended range. The VAX results presented here all used the D – floating option for higher precision. The MacFortran/020 single precision also claims 7 -digit precision[3] and double precision claims 15-digit precision. The coprocessor carries calculations and intermediate results to 80-bit precision.

Macintosh provides its own numeric environment called SANE[4] that implements the IEEE 754 floating point arithmetic standards. SANE provides a 15-bit exponent and a 64-bit mantissa. Even though the MacFortran/020 does not use the SANE environment, it does conform to the IEEE 754 standard. The Macintosh 68881 math coprocessor option was used in MacFortran.

Table II summarizes the different single and double precision floating point representations for the VAX Fortran and MacFortran/020 systems. In the following comparisons, results from a CRAY-YMP using CFT77 version 3.0, are used as the standard for comparing both the single precision and the double precision real data type results. Since CFT77 does not support a double precision complex data type, CRAY results were not used in the double precision complex comparisons.

IV. SINGLE PRECISION RESULTS

Two additional computer systems are included for comparative purposes. These are a SUN SPARC workstation, with OS version 4.0.3, and a Silicon Graphics Inc. (SGI), IRIS-4D-220 minicomputer, with OS IRIX 3.2.1. Real and complex output from the modified covariance method from these systems are compared with the output from the CRAY. Although not shown, the results from a Compaq 386 PC using Microsoft Fortran 4.1 agreed exactly with the MacFortran/020. In the following discussion, the relative error with respect to the CRAY output are reported as results. The relative error is determined by the difference in output from the CRAY and the other computer system, then the difference is divided by the CRAY result, (|CRAY—Other system|)/CRAY. Thus, if the absolute

TABLE II. Comparison of FORTRAN floating point representation.

System	Mantissa bits	Exponent bits
CRAY-YMP single precision	47	15
VAX single precision	23	8
VAX D – floating double precision	55	8
VAX G – floating double precision	52	11
MAC single precision	23	8
MAC double precision	52	11

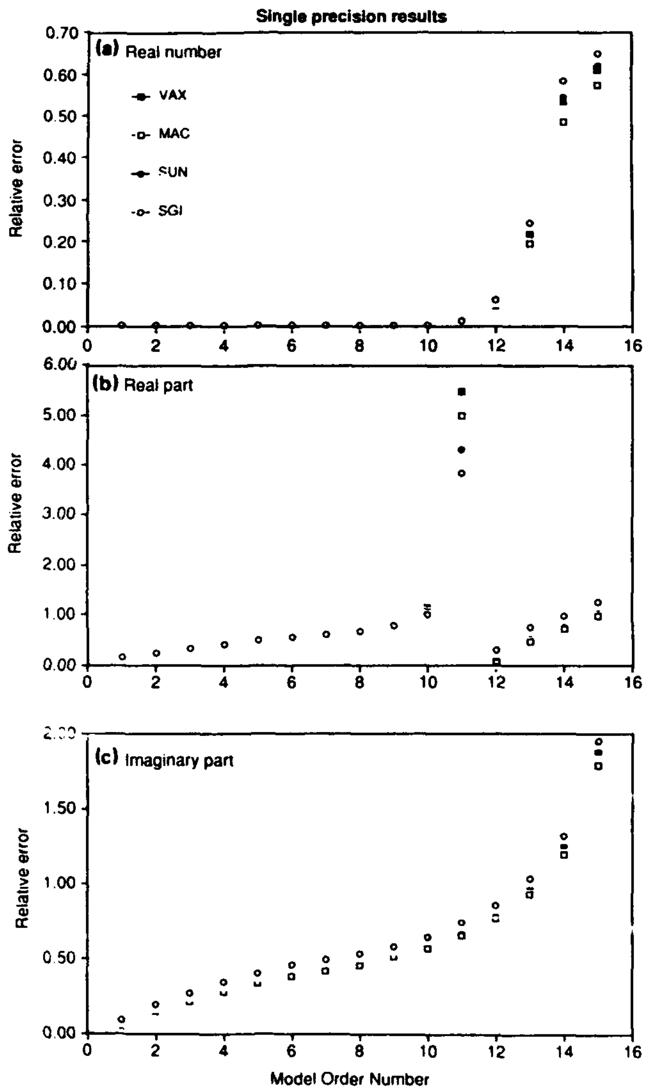
difference is small, and hence the relative error is small, then the result from that computer system agrees with the CRAY result. Relative error is more magnitude independent than differences. However, since the numbers being compared should be nearly equal, the relative error can be deceptively severe. Therefore, the following plots show relative error and differences.

The single precision results are shown in Fig. 1. The relative error for the real data type is shown in Fig. 1(a) as a function of model order number which represents the number of recursions used to obtain results. The relative error is negligible out to model order ten for all four computer systems. Above ten relative error increases dramatically up to 60%. The differences (Cray—other system), although not shown, show the same trend. Differences in the second decimal place begin at model order 1 and, by model order 13, differences occur in the first decimal place.

The differences between the VAX, MAC, SUN, and SGI results are small compared to the difference of any of these systems with the CRAY due to the deficiency of 24-bit computations versus 48-bit computations. Because of this and the large span of differences, the graphs tend to overlap each other. For lower model orders, only the last plotted symbol is visible as it plots over the others. By model order 12 the differences are sufficient to be individually discernible.

Figure 1(b) shows the relative error for the real part of a complex number and Fig. 1(c), the imaginary part. The real part shows a linear increase in relative error to model order 10, then a large error at model order 11, followed by a smaller errors with a larger linear slope. The imaginary part, Fig. 1(c), shows a smooth but increasing error out to model order 10, then larger errors. The relative errors for the real part are noticeably different from the imaginary part.

In terms of differences, the real part of the complex number, Fig. 1(d), starts near zero, peaks at model order 6, and returns to near zero. The differences in the imaginary part, Fig. 1(e), appear to be the mirrored reflection of the real part. Starting near zero, the differences dip and return



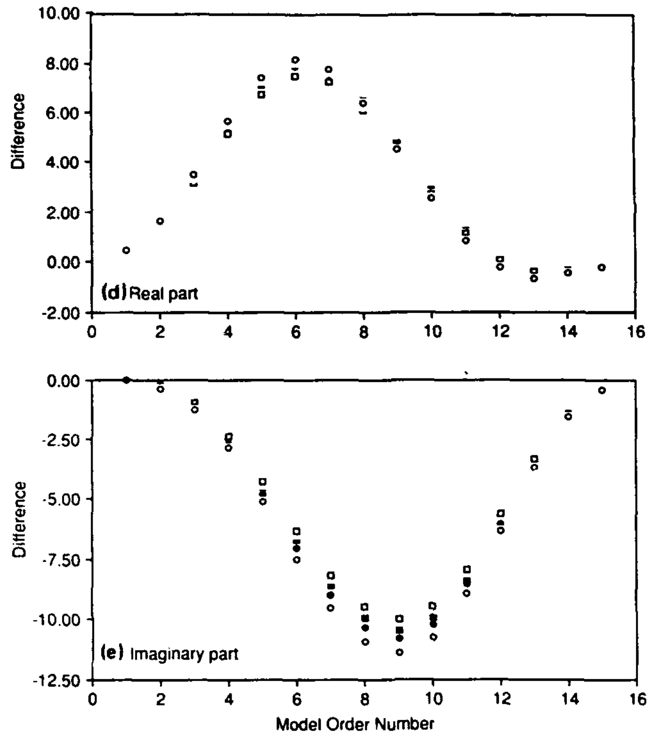


FIG. 1. Comparisons of single precision FORTRAN results with respect to the CRAY from four computer systems for two data types. (a) Real data type relative error; (b) real part of a complex number relative error; (c) imaginary part of the complex number relative error; (d) real part of a complex number difference; and (e) imaginary part of the complex number difference. Because the differences are small among the four computers are small, the plotting symbols tend to overlap.

TABLE III. Single precision results averaged for 15 model orders.

System	Real number	Complex number Real	Complex number Imaginary
	Average difference (CRAY—other system)		
VAX	− 0.02	3.21	− 4.91
MAC	- 0.02	3.02	− 4.67
SUN	− 0.02	3.10	− 5.09
SGI	− 0.03	3.16	− 5.39
	Average relative error		
VAX	0.09	0.86	1.88
MAC	0.09	0.81	1.79
SUN	0.10	0.81	1.94
SGI	0.10	0.82	2.05

to zero. As a function of model order, when differences occur in the real part, they also occur in the imaginary part. It is not clear whether the difference dependence on model order is primarily a result of numeric errors or the input data used in the method. The differences do not show the large error at model order 10 that the relative error shows, indicating that numeric errors associated with the division in computing relative error may be the cause of error at this particular model order. It is interesting to note that this occurs only for the real part of the complex number.

Table III gives the average difference with respect to the CRAY result by averaging over model order. Averaging determines which system gives better overall agreement with the CRAY result. All four computer systems match the CRAY result to the same order of magnitude.

V. DOUBLE PRECISION RESULTS

Figure 2 is similar to Fig. 1 but shows double precision results with respect to the VAX using a D − floating option. The relative errors are given in Fig. 1(a), for the real number, Fig. 1(b) for the real part of the complex number, and Fig. 1(c) for the imaginary part of the complex number. The differences with respect to the VAX output are shown in Fig. 1(d) for the real part of the complex number, and Fig. 1(e) for the imaginary part of the complex number.

Figure 2(a) shows the relative error with respect to the VAX for the real number data type. The relative error is

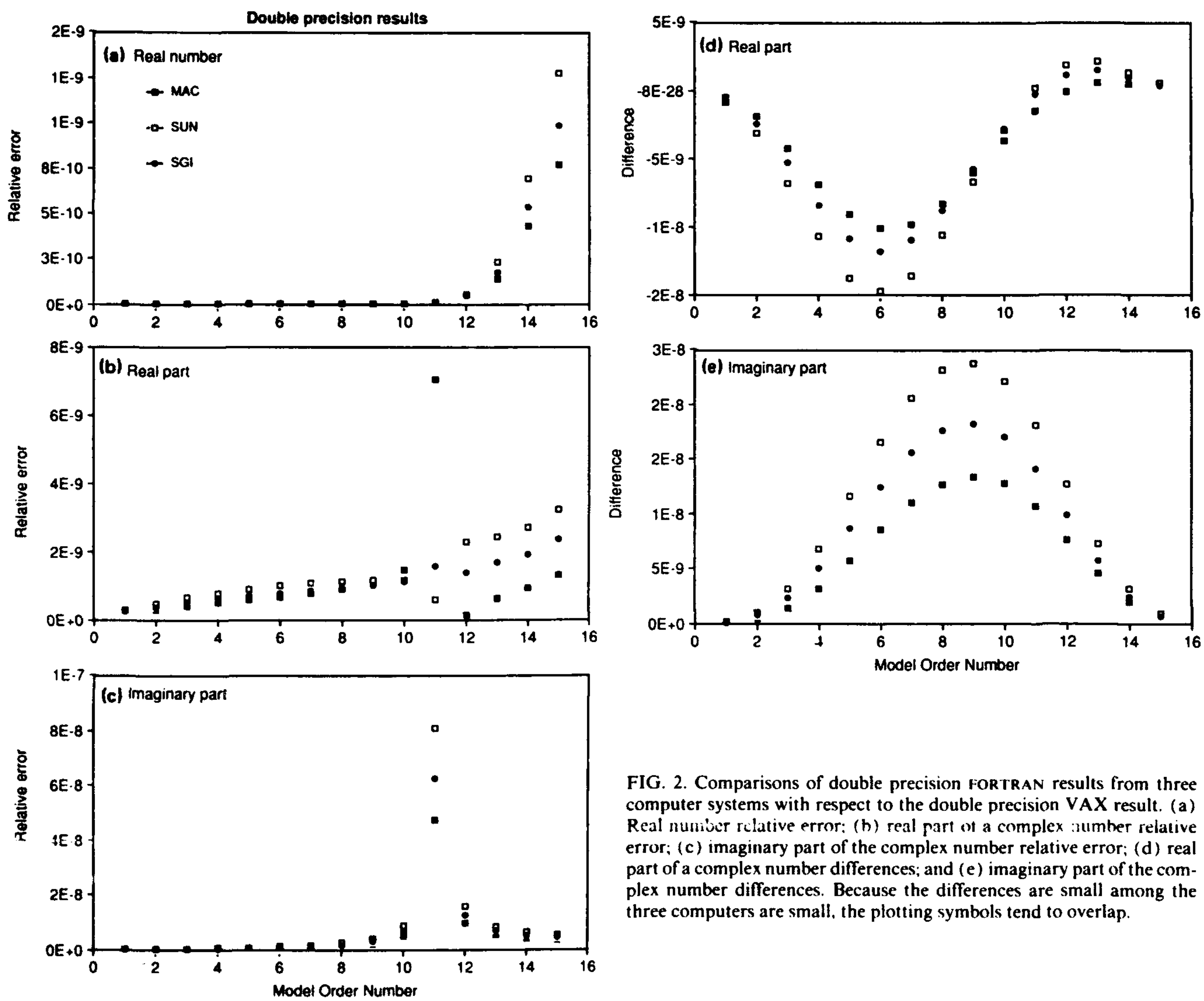


FIG. 2. Comparisons of double precision FORTRAN results from three computer systems with respect to the double precision VAX result. (a) Real number relative error; (b) real part of a complex number relative error; (c) imaginary part of the complex number relative error; (d) real part of a complex number differences; and (e) imaginary part of the complex number differences. Because the differences are small among the three computers are small, the plotting symbols tend to overlap.

TABLE IV. Double precision results averaged for 15 model orders.

System	Average difference (VAX—other system) Real number	Complex number Real	Imaginary
MAC	3.132E-11	− 4.106E-9	6.292E-9
SUN	5.138E-11	− 5.222E-9	1.140E-8
SGI	3.973E-11	− 4.318E-9	8.714E-9
	Average relative error		
MAC	9.191E-11	1.103E-9	2.404E-9
SUN	1.509E-10	1.243E-9	4.269E-9
SGI	1.168E-10	1.064E-9	3.274E-9

negligible out to model order 10 for all three computer systems, then the relative error increases dramatically for the higher model orders. The differences, although not shown, do show the same trend. For model orders less than 11, the double precision real number differences are zero to 13 decimal places. For model orders greater than 12, differences begin to increase.

Figure 2(b) is the relative error for the real part of the complex number and Fig. 2(c) the imaginary part. For the real part, the relati e error increases linearly up to model order 10. After model order 10, the differences among the other computer systems is more easily discernible and exhibit a general agreement with the single precision real part results. The imaginary part is different than the single precision counterpart in that it exhibits some large error for model order 11. The differences in Fig. 2(d) and (e) do not show this, so this spike is believed to be a numeric division problem in computing relative error.

Differences for the real part of the complex number, Fig. 2(d), start out near zero, dip, then return to near zero. As in the single precision case, the differences in the imaginary part, Fig. 2(e), appear to mirror the real part in both magnitude and model order: starting near zero, peaking, and returning to zero. The dependence on model order for

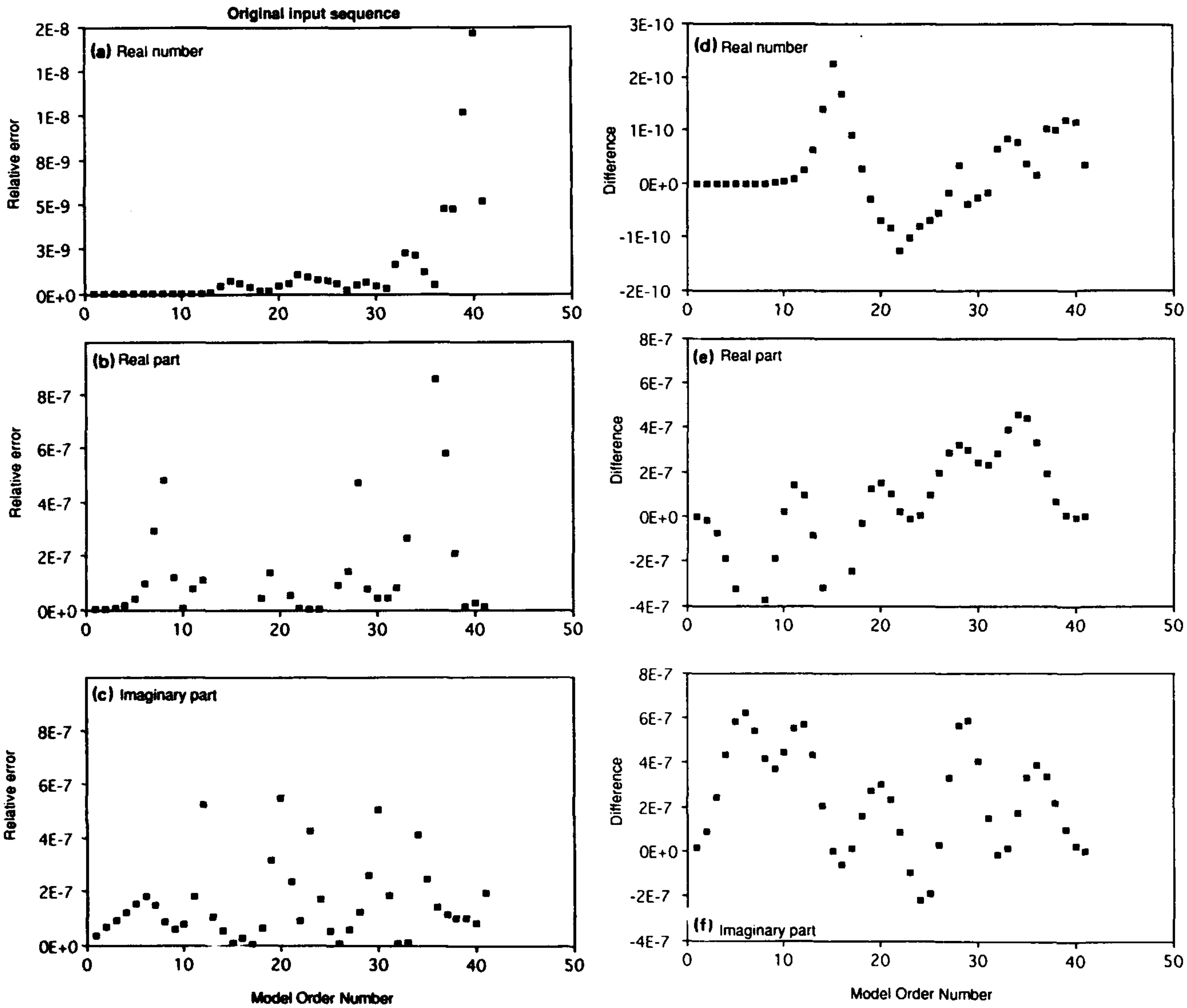


FIG. 3. Comparisons of double precision relative errors and differences for the VAX Fortran 4.8 and MacFortran/020 for higher model orders. (a) Real data type; (b) real part of a complex number; (c) imaginary part of the complex number; (d) the relative difference for the real number; (e) relative difference for the real part of a complex number; and (f) relative difference for the imaginary part of the complex number.

the imaginary part looks like the mirrored reflection of the real part.

Table IV compares the double precision relative error and differences for each computer system averaged over all 15 recursion steps. Table IV shows that the VAX floating point representation, although having three more bits in the mantissa does not provide significantly better results than the other systems. The double precision differences among the computers are small.

VI. LARGE MODEL ORDER RESULTS

To check consistency in the differences between the MacFortran/020 and VAX Fortran 4.8, and to see how large these differences might grow, another case was examined. This case used the same input data, full double precision, and a higher model order, 41 instead of 15. This gives over 13 000 total multiplications and additions. To compare double precision relative errors and differences at higher model orders, the VAX is used as the standard. Figure 3 shows the relative errors and differences for both real and complex data types.

For the real data type, Fig. 3(a), the relative error shows two local maxima near model order 15 and 22, but is small out to model order 30. Above model order 30 the error grows. It is noted that similar differences with respect to the Cray result also occurred near model order 15 [Fig. 2(a)]. The complex data type clearly shows less agreement between the VAX and Mac result than the real data type. The real part shows some severe relative errors, one near model order 9, one is offscale at model order 15, and near 28 and 35. The imaginary part also has a spikey nature with large relative errors near model order 12, 20, 22, 30, and 35. The real data type shows some disagreement between the relative error, Fig. 3(a), and the relative difference, Fig. 3(d). When the difference is large, near model order 15 and 22, the relative error is small. When the relative error is large, near model order 40, the difference is small. The complex number shows better agreement between the relative error, Fig. 3(b) and (c) and the relative difference, Fig. 3(e) and (f).

In terms of differences, the VAX and MacFortran/020 agree out to ten decimal places up to model order number 12. Both the real and imaginary parts show differences in the seventh decimal place for model orders less than 10; however, the imaginary part, once again, seems to mirror the real part. These double precision results suggest that complex data types have less arithmetic precision for than real data types.

In an attempt to eliminate the effect of input data, the sequence of the input data was reversed and the program executed with a model order 41. This relative errors are shown in Fig. 4 and the magnitudes are basically the same. However, the directions are reversed, when one ordering causes a positive difference, the other ordering gives a negative difference. Thus the input data sequencing does not play an important role in the level of accuracy and precision of these results.

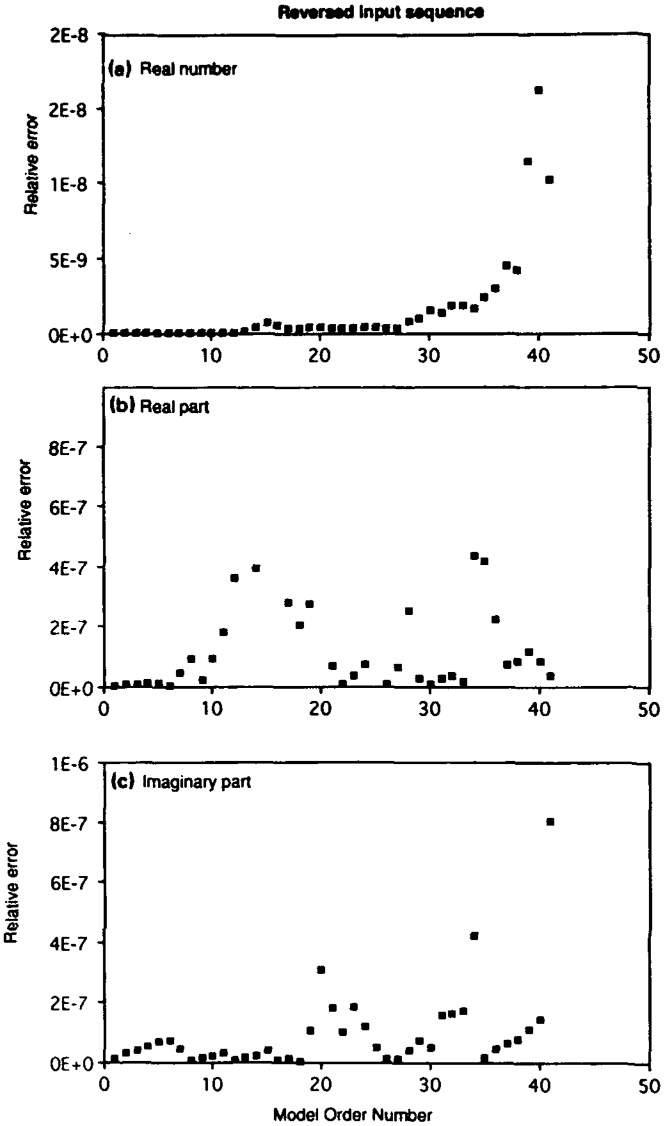


FIG. 4. Comparisons of double precision relative errors with the input data reversed. (a) Real data type; (b) real part of a complex number; and (c) imaginary part of the complex number.

VII. CONCLUSIONS

Small differences occurred in numeric results from various computer systems. The CRAY results were used as the standard for single precision comparisons and the VAX for double precision comparisons. It is impossible to assign differences reported here solely to compiler differences. The roles of the hardware, input data conversions, and the recursion method are difficult to separate. It was shown that sometimes computer codes need more extensive testing than simply obtaining the published answers. In this example, *double precision implementation is required to* achieve satisfactory agreement between different computer systems out to large model orders.

Arithmetic with complex data types, especially for large computations, is shown to be less precise than for real data types. VAX Fortran 4.8, MacFortran/020, SUN, and SGI results all show the same precision with respect to CRAY results. The differences among them are an order of magnitude smaller compared to the difference with the CRAY results. In this example, MacFortran/020 single precision is slightly closer to CRAY results than the VAX Fortran 4.8 and VAX Fortran double precision with the D – floating option is slightly closer to CRAY results than MacFortran/020.

From the user point of view, this algorithm requires a double precision implementation. Fig. 1 and Table III show first digit differences in single precision results for the book example at model order 15. When using the modified covariance method, the numerical errors are dependent on the model order chosen. These errors are not linearly dependent on model order. Complex data type differences are larger than the real data type. There is a relationship between the errors in the real and imaginary parts of the complex number. This may be a result of manipulating complex numbers which usually involves converting complex pairs to the polar coordinate system in order to facilitate multiplication and addition, and hence uses transcendental functions. The use of transcendental functions would add to numerical errors and may be the cause of the phase like errors between the real and imaginary component differences.

ACKNOWLEDGMENTS

The author gratefully acknowledges the help and assistance of Bill Maloy, Guy Norton, Keith Davis, and Al Leybourne with the various computer systems. This work was sponsored by the Office of Naval Technology under the management of the Naval Oceanographic and Atmospheric Research Laboratory. The mention of commercial products or use of company names does not in any way imply endorsement by the U.S. Navy or NRL-SSC.

REFERENCES

1. S. Lawrence Marple, Jr., *Digital Spectral Analysis with Applications* (Prentice-Hall, Englewood Cliffs, NJ, 1987).
2. Digital Equipment Corporation, "VAX-11 Fortran Reference Manual," December 1987, pp. 2-5 through 2-10 of VAX Fortran manual.
3. "MacFortran/020 Reference Manual," Absoft Corporation (1987).
4. Sazegari, Ali, "Macintosh Numerics: An environment for Scientific Computing," Comput. Phys. 4 (4) (1990).